1. (Currently Amended) A method for identifying presence of malicious code in program code within a computer system, the method comprising:

initializing a virtual machine within the computer system, the virtual machine comprising a virtual personal computer (PC) implemented by software simulating functionality of a central processing unit and memory and a virtual operating system simulating functionality of a multi-threaded operating system of the computer system;

virtually executing a target program within the virtual PC so that the target program interacts only with an instance of the virtual operating system;

analyzing behavior of the target program upon completion of virtual execution to identify an occurrence of malicious code behavior based upon an evaluation by the virtual machine of a behavior pattern representing information about all functions simulated by the target program during virtual execution;

generating the behavior pattern for the target program by tracking functions performed and not performed by the target program with flags in a behavior pattern field and by tracking a sequence in which the functions are called by the target program with the behavior pattern field; and

terminating the virtual PC after the analyzing process, thereby removing from the computer system a copy of the target program that was contained within the virtual PC.

2. (Previously Presented) The method of claim 1, wherein the virtual PC of the virtual machine simulates functionality of input/output ports, and the virtual operating system simulates functionality of operating system data areas and an operating system application program interface.

3. (Previously Presented) The method of claim 1, wherein the virtual operating system is operative to simulate an application program interface call of the operating system by returning a correct value to the call without completing actual performance of the call.

4. (Original) The method of claim 2, wherein virtual execution of the target program causes the target program to interact with the simulated operating system application program interface.

5. (Previously Presented) The method of claim 1, wherein the target program is newly introduced to the computer system and initially executed by virtually executing the target program on the virtual PC.

6. (Previously Presented) The method of claim 1, wherein after a first instance of the target program is analyzed by the virtual machine and a first behavior pattern is generated and stored in a database coupled to the computer system, the method further comprising:

determining that the target program is modified;

analyzing the modified target program by executing the modified target program in the virtual machine to provide a second behavior pattern; and

comparing within the virtual machine the first behavior pattern to the second behavior pattern to determine whether the second behavior pattern is altered from the first behavior pattern in a manner indicative of presence of the malicious code in the modified target program.

7. (Previously Presented) The method of claim 6, wherein a new behavior pattern is generated each time the target program is modified.

8. (Previously Presented) The method of claim 6, wherein introduction of the malicious code during modification of the target program is detected by comparing the first behavior pattern to the second behavior pattern and identifying altered bits indicating an addition of an infection procedure to the modified target program.

9. (Previously Presented) The method of claim 6, wherein the first behavior pattern is identified as a match to the second behavior pattern when the modified target program is a new version of the first program.

10. (Previously Presented) The method of claim 1, wherein the behavior pattern identifies functions executed in the virtual execution of the target program, the method further comprising tracking an order in which the functions are virtually executed by the target program within the virtual PC to provide a complete record of all functions simulated by the target program, as if the target program were executed on the computer system.

[The remainder of this page has been intentionally left blank.]

11. (Currently Amended) A method for identifying presence of malicious code in program code within a computer system, the method comprising:

initializing a virtual machine within the computer system, the virtual machine comprising software simulating functionality of a central processing unit and memory and a virtual operating system simulating functionality of a multi-threaded operating system of the computer system;

virtually executing a target program with the virtual machine so that the target program interacts with an instance of the virtual operating system rather than with the operating system of the computer system, whereby the malicious code is fully executed during virtual execution of the target program if the target program comprises the malicious code;

generating a behavior pattern for the target program by tracking functions performed and not performed by the target program with flags in a behavior pattern field and by tracking a sequence in which the functions are called by the target program with the behavior pattern field in order to collect information about all functions simulated by the target program during virtual execution; and

terminating the virtual machine upon completion of the virtual execution of the target program, leaving behind a record of the behavior pattern that is representative of operations of the target program with the computer system, including operations of the malicious code if the target program comprises the malicious code.

12. (Original) The method of claim 11, wherein the record is in a behavior register in the computer system.

[The remainder of this page has been intentionally left blank.]

13. (Previously Presented) The method of claim 11, wherein after a first instance of the target program is analyzed by the virtual machine and a first behavior pattern is generated and stored in a database coupled to the computer system, the method further comprising:

determining that the target program is modified;

analyzing the modified target program by executing the modified target program with the virtual machine to provide a second behavior pattern; and

comparing the first behavior pattern to the second behavior pattern to determine whether the second behavior pattern is altered from the first behavior pattern in a manner indicative of presence of the malicious code in the modified target program.

14. (Previously Presented) The method of claim 13, wherein a new behavior pattern is generated each time the target program is modified.

15. (Previously Presented) The method of claim 13, wherein introduction of the malicious code during modification of the target program is detected by comparing the first behavior pattern to the second behavior pattern and identifying altered bits indicating an addition of an infection procedure to the modified target program.

16. (Previously Presented) The method of claim 13, wherein the first behavior pattern is identified as a match to the second behavior pattern when the modified target program is a new version of the first program.

17. (Previously Presented) The method of claim 11, wherein the behavior pattern identifies all functions executed during the virtual execution of the target program and records an order of simulation of the functions.

18. (Currently Amended) A memory storage device comprising computer-executable steps for identifying the presence of malicious code in program code in a computer system, comprising:

initializing a virtual machine for the computer system, the virtual machine comprising a virtual personal computer (PC) implemented by software simulating functionality of a central processing unit memory a virtual operating system simulating functionality of a multi-threaded operating system of the computer system;

executing a target program within the virtual PC so that the target program completes a virtual execution by interacting only with an instance of the virtual operating system;

generating a behavior pattern by completing virtual execution of the target program within the virtual PC and by tracking functions performed and not performed by the target program with flags in a behavior pattern field and by tracking a sequence in which the functions are called by the target program, the behavior pattern representative of operational functions completed by the target program during virtual execution, including at least one of virtual operating system calls, Input/Output functions and program functions supported by the target program;

upon completion of virtual execution, operating the virtual machine to compare the behavior pattern generated by virtual execution of the target program to a behavior pattern representative of operations by the malicious code to identify an occurrence of malicious code behavior; and

in the event that the comparison process results in a match representing an identification of malicious code behavior by the target program, then identifying the target program as comprising the malicious code.

19. (Currently Amended) The memory storage device of Claim 18 further comprising the computer-executable step of removing the target program from the computer system in response to an identification of the target program comprising malicious code so that the target program cannot affect the performance of subsequent programs executed by the computer system.

20. (Currently Amended) A memory storage device comprising computer-executable steps for identifying the presence of malicious code in program code in a computer system, comprising:

executing a target program within a virtual personal computer (PC) so that the target program completes a virtual execution by interacting only with an instance of a virtual operating system, the virtual PC comprising software operative to simulate functionality of a processor and memory, the virtual operating system operative to simulate functionality of a multi-threaded operating system for the computer system, the virtual PC and the virtual operating system operating in combination to form a virtual machine;

collecting information about the behavior of the target program during virtual execution of the target program by the virtual machine by tracking functions performed and not performed by the target program with flags in a behavior pattern field and by tracking a sequence in which the functions are called by the target program in order to create a record of virtual operations of the target program, whereby the record reflects a plurality of operations of the malicious code if the target program comprises the malicious code;

upon completion of virtual execution of the target program, analyzing the record with the virtual machine to identify an occurrence of malicious code behavior by comparing the record to a behavior pattern representative of the operations performed by the malicious code; and

in the event that the record matches the malicious code behavior, then identifying the target program as comprising the malicious code.

21. (Currently Amended) The memory storage device of Claim 20 further comprising the computer-executable step of removing the target program from the computer system in response to an identification of the target program comprising malicious code so that the target program cannot affect performance of subsequent programs executed by the computer system.

22. (Currently Amended) A computer-implemented method for identifying a presence of malicious code in program code for a computer system, comprising the steps:

virtually executing a target program within a virtual machine comprising a virtual personal computer (PC) implemented by software operative to simulate functionality of a processor, and memory and a virtual operating system having software simulating functionality of a multi-threaded operating system for the computer system wherein virtual execution of the target program comprises interactions with an instance of the virtual operating system; [[and]]

creating a record of all functions simulated by the target program during virtual execution of the target program by the virtual machine, the record comprising a behavior pattern representative of the behavior of the target program as if it were executed on the computer system, the behavior pattern comprising characteristics of malicious code behavior in the event that the target program comprises the malicious code; and

creating the behavior pattern by tracking functions performed and not performed by the target program with flags in a behavior pattern field and by tracking a sequence in which the functions are called by the target program with the behavior pattern field.

23. (Previously Presented) The computer-implemented method of Claim 22 further comprising the step of operating the virtual machine to analyze the record after completion of the virtual execution by the target program to identify an occurrence of a type of the behavior pattern representative of operations by the malicious code.

24. (Previously Presented) The computer-implemented method of Claim 23 wherein, in the event of an identification of an occurrence of malicious code behavior by the target program, the method further comprises the step of identifying the target program as comprising the malicious code.

25. (Currently Amended) The computer-implemented method of Claim 24 further comprising the step of removing the target program from the computer system in response to an identification that the target program comprises the malicious code so that the target program cannot affect performance of subsequent programs executed by the computer system.

26. (Currently Amended) A memory storage device comprising computer-executable steps for identifying the presence of malicious code in program code in a computer system, comprising:

executing a target program within a virtual personal computer (PC) so that the target program completes a virtual execution by interacting only with an instance of a virtual operating system, the virtual PC comprising software operative to simulate functionality of a processor and memory, the virtual operating system operative to simulate functionality of a multi-threaded operating system for the computer system, the virtual PC and the virtual operating system operating in combination to form a virtual machine;

collecting information about the behavior of the target program in response to virtual execution of the target program by the virtual machine;

in response to completing virtual execution of the target program, collecting information about interrupt call operations that call any interrupt service routine modified by the virtual execution of the target program;

creating a record by tracking functions performed and not performed by the target program with flags in a behavior pattern field and by tracking a sequence in which the functions are called by the target program with the behavior pattern field, the functions comprising the interrupt call operations, the record comprising the information collected about the virtual execution of the target program and the interrupt call operations that call any interrupt service routine modified by the virtual execution of the target program;

analyzing the record to identify an occurrence of malicious code behavior by comparing the record to a behavior pattern representative of the operations performed by the malicious code; and

in the event that the record matches the malicious code behavior, then identifying the target program as comprising the malicious code.

27. (Currently Amended) The memory storage device of Claim 26 further comprising the computer-executable step of removing the target program from the computer system in response to an identification that the target program comprises malicious code so that the target program cannot affect performance of subsequent programs executed by the computer system.

28. (Currently Amended) The memory storage device of Claim 26, wherein the step of collecting information about the behavior of the target program in response to virtual execution of the target program comprises storing bits that correspond to the flags in a behavior pattern register, the behavior pattern register providing memory for the behavior pattern field, the storing of the bits being completed in response to monitoring operating system calls, interrupts and I/O port read/write operations completed by the virtual machine.

[The remainder of this page has been intentionally left blank.]

29. (Currently Amended) A memory storage device comprising computer-executable steps for identifying the presence of malicious code in program code in a computer system, comprising:

initializing a virtual machine for the computer system, the virtual machine comprising a virtual personal computer (PC) implemented by software simulating functionality of a central processing unit and memory and a virtual operating system simulating functionality of a multi-threaded operating system of the computer system,

the initializing step comprising the steps of extracting the file structure of a target program and loading the target program into the software-simulated memory of the virtual PC;

executing a target program within the virtual PC so that the target program completes a virtual execution by interacting only with an instance of the virtual operating system;

generating a behavior pattern by completing virtual execution of the entire code of the target program within the virtual PC and by tracking functions performed and not performed by the target program with flags in a behavior pattern field and by tracking a sequence in which the functions are called by the target program, the behavior pattern representative of a sequence of operational functions completed by the target program during virtual execution, including at least one of virtual operating system calls, Input/Output functions and program functions supported by the target program;

upon completion of virtual execution, operating the virtual machine to compare the behavior pattern generated by virtual execution of the target program to a behavior pattern representative of operations by the malicious code to identify an occurrence of malicious code behavior; and

in the event that the comparison process results in a match representing an identification of malicious code behavior by the target program, then identifying the target program as comprising the malicious code.

30. (Currently Amended) The memory storage device of Claim 29 further comprising the computer-executable step of removing the target program from the computer system in response to an identification that the target program comprises malicious code so that the target program cannot affect performance of subsequent programs executed by the computer system.

31. (Previously Presented) The memory storage device of claim 29, further comprising the computer-executable steps of:

identifying a new instance of the target program;

determining that the new instance of the target program represents a modified version of the target program;

analyzing the modified target program by executing the modified version of the target program in the virtual machine to provide a supplemental behavior pattern; and

comparing within the virtual machine the behavior pattern to the supplemental behavior pattern to determine whether the supplemental behavior pattern is altered from the behavior pattern in a manner indicative of presence of the malicious code in the modified version of the target program.

32. (Previously Presented) The memory storage device of claim 29, wherein another supplemental behavior pattern is generated each time the target program is modified.

33. (Previously Presented) The memory storage device of claim 29, wherein the malicious code is detected by comparing the behavior pattern to the supplemental behavior pattern and identifying altered bits indicating an addition of an infection procedure to the modified version of the target program.

34. (Previously Presented) The memory storage device of claim 29, wherein the behavior pattern is identified as a match to the supplemental behavior pattern when the modified version of the target program is a new version of the first program.